

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A method for a computer system for providing communication between a first system and a second system, wherein the first system uses IPv4 (Internet Protocol version 4) ~~a first version of a protocol~~ and the second system uses IPv6 (Internet Protocol version 6) ~~a second version of the protocol~~, the method comprising:
 - providing a first application on the first system;
 - inserting an Application Programming Interface (API) ~~API~~ level translator layer, wherein the API level translator layer does not require a stack for IPv6 ~~the second version of the protocol~~;
 - making a function call to a socket ~~Application Programming Interface (API)~~ for IPv4 ~~the first version~~;
 - intercepting the function call by the API level translator layer, wherein interception of the function call is not required to occur after a packet is passed to an IPv4 stack;
 - translating the function call to a translated function call wherein the translated function call uses raw sockets;
 - making a function call to the socket API for the translated function call that uses raw sockets; ~~and~~
 - setting the socket option on the raw socket to supply new IP headers;
 - passing a packet to a stack for IPv4 ~~the first version of the protocol~~;
 - providing an alternate implementation for a sending-related IPv4 socket function, wherein the alternate implementation comprises:
 - intercepting an IPv4 socket API call to send a packet;
 - translating the IPv4 socket API call to use a raw socket;

providing transport and IPv6 headers ahead of user supplied payload data;
fragmenting the packet as required;
calling a corresponding IPv4 socket API function for the raw socket; and
passing the packet to the stack; and
providing an alternate implementation for a reception-related IPv4 socket function,
wherein the alternate implementation comprises:
receiving an incoming packet on a raw socket;
checking a source host to determine the proper destination for the
incoming packet as required;
checking a port number for the incoming packet;
stripping transport and IP headers from the incoming packet; and
passing a payload to a destination application.

2. (Canceled)
3. (Canceled)
4. (Currently Amended) The method of claim 1 2, further comprising supplying IP headers only once.
5. (Currently Amended) The method of claim 1 2, wherein the method is implemented by inserting an API level translator layer between a socket API layer and a TCP/IPv4 layer.
6. (Currently Amended) The method of claim 1 2, wherein the method is implemented without using an IPv6 stack.

7. (Original) The method of claim 4, further comprising passing the packet to a network card driver.
8. (Canceled)
9. (Currently Amended) The method of claim 1 ~~2~~, further comprising performing tunneling of IPv6 packets over IPv4 routing infrastructure.
10. (Currently Amended) The method of claim 1 ~~2~~, further comprising using a Name Resolver service to perform name to address resolution related functions.
11. (Original) The method of claim 10, wherein the Name Resolver service is configured to run on a separate host that includes an IPv4 stack and an IPv6 stack.
12. (Original) The method of claim 11, wherein name to address resolution functions of the Name Resolver service use the IPv6 stack.
13. (Original) The method of claim 12, wherein the Name Resolver service is further configured to receive a query from the first system, use the address resolution functions to obtain a record and send the record to the first system.
14. (Canceled)
15. (Canceled)
16. (Currently Amended) The method of claim 1 ~~14~~, further comprising fragmenting the packet.

17. (Currently Amended) The method of claim 14, further comprising passing the packet to a network card driver.

18. (Currently Amended) A set of executable instructions on a computer readable medium for providing communication between an IPv4 (Internet Protocol version 4) system and an IPv6 (Internet Protocol version 6) system, the instructions being configured to:

provide an IPv4 application on the IPv4 system;

insert an Application Programming Interface (API) API level translator layer, wherein the API level translator layer does not require a stack for IPv6 ~~the second version of the protocol~~;

make a function call to an IPv4 socket ~~Application Programming Interface (API)~~;

translate the function call to a translated function call wherein the translated function call uses raw sockets;

make another function call to the IPv4 socket API for the translated function call that uses raw sockets; ~~and~~

set the socket option on the raw socket to supply new IP headers;

pass a packet to an IPv4 stack;

provide an alternate implementation for a sending-related IPv4 socket function, wherein the alternate implementation comprises:

intercepting an IPv4 socket API call to send a packet;

translating the IPv4 socket API call to use a raw socket;

providing transport and IPv6 headers ahead of user supplied payload data;

fragmenting the packet as required;

calling a corresponding IPv4 socket API function for the raw socket; and

passing the packet to the stack; and

provide an alternate implementation for a reception-related IPv4 socket function, wherein

the alternate implementation comprises:

receiving an incoming packet on a raw socket;

checking a source host to determine the proper destination for the
incoming packet as required;

checking a port number for the incoming packet;

stripping transport and IP headers from the incoming packet; and
passing a payload to the IPv4 application.

19. (Original) The set of executable instructions of claim 18, wherein the method further comprises supplying IP headers only once.

20. (Original) The set of executable instructions of claim 18, wherein the method is implemented by inserting an API level translator layer between a socket API layer and a TCP/IPv4 layer.

21. (Original) The set of executable instructions of claim 18, wherein the method is implemented without using an IPv6 stack.

22. (Canceled)

23. (Canceled)

24. (Canceled)

25. (Currently Amended) The set of executable instructions of claim 18 ~~24~~, wherein the method further comprises performing tunneling of IPv6 packets over IPv4 routing infrastructure.

26. (Currently Amended) The set of executable instructions of claim 18 ~~24~~, wherein the method further comprises fragmenting the packet.

27. (Currently Amended) The set of executable instructions of claim 18 ~~24~~, wherein the method further comprises passing the packet to a network card driver.

28. (Currently Amended) A system for enabling an IPv4 (Internet Protocol version 4) application to communicate across a computer network using an IPv6 (Internet Protocol version 6) system, the system comprising:

a computing device;

executable instructions executable on the computing device, wherein the executable instructions are configured ~~to implement a method comprising:~~

insert ~~inserting~~ an Application Programming Interface (API) ~~API~~ level translator layer, wherein the API level translator layer does not require a stack for IPv6 ~~the second version of the protocol~~;

make ~~making~~ a function call to an IPv4 socket ~~Application Programming Interface (API)~~;

translate ~~translating~~ the function call to a translated function call wherein the translated function call uses raw sockets;

make ~~making~~ another function call to the IPv4 socket API for the translated function call that uses raw sockets; ~~and~~

set the socket option on the raw socket to supply new IP headers;

pass ~~passing~~ a packet to an IPv4 stack;

provide an alternate implementation for a sending-related IPv4 socket function,

wherein the alternate implementation comprises:

intercepting an IPv4 socket API call to send a packet;

translating the IPv4 socket API call to use a raw socket;
providing transport and IPv6 headers ahead of user supplied payload data;
fragmenting the packet as required;
calling a corresponding IPv4 socket API function for the raw socket; and
passing the packet to the stack; and
provide an alternate implementation for a reception-related IPv4 socket function,
wherein the alternate implementation comprises:
receiving an incoming packet on a raw socket;
checking a source host to determine the proper destination for the
incoming packet as required;
checking a port number for the incoming packet;
stripping transport and IP headers from the incoming packet; and
passing a payload to the IPv4 application.

29. (Original) The system of claim 28, further comprising an API level translator layer between a socket API layer and a TCP/IPv4 layer.

30. (Canceled)

31. (Canceled)

32. (Original) The system of claim 28, wherein the method further comprises performing tunneling of IPv6 packets over IPv4 routing infrastructure.

33. (Original) The system of claim 28, wherein the method further comprises fragmenting the packet.